

REMARKS

Claims 14-33 are pending in the present patent application. The Examiner has rejected claims 14-33. Applicant has amended claims 14-17, 20-22, and 27-28. Applicant respectfully requests reconsideration of claims 14-33 in view of at least the following amendments and remarks.

I. Claim objections

Examiner has objected to claim 28 stating:

Claim 28 is objected to because of the following informalities: --a—should be inserted between “comprising” and “first” in line 6. Appropriate correction is required.

In response, Applicant has amended claim 28 as suggested by Examiner.

II. Rejection of Claims 14-20 Based on 35 U.S.C. § 112

Examiner has rejected claims 14-20 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Examiner states:

Claim 14 is unclear because “computer readable program code” in line 6 seems out of context.

Claim 14 recites the limitation “said control program code” in line 7. There is insufficient antecedent basis for this limitation in the claim.

Claim 14 recites the limitation “said computer readable control program code” in line 8. There is insufficient antecedent basis for this limitation in the claim.

Claim 20 is indefinite because “configured to generate definitional statement” in lines 6-7 is ambiguous. It is not clear if there should be singular or plural definitional statements.

In response, Applicant has amended claims 14-20 to eliminate any ambiguities as to the claimed invention.

III. Rejection of Claims 14-18 and 20-32 Based on 35 U.S.C. § 102(e)

Examiner has rejected claims 14-18 and 20-32 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,249,291 to Popp et al. (hereinafter Popp). Examiner states:

Referring to claim 14, Popp discloses a method in a computer system and a computer usable medium having computer readable program code that uses a namespace in generating a GUI (web pages in a browser). See the description about Group Object on page 15, line 36- page 16, line 47 for description of how Popp uses namespaces. A Name property identifies the group (namespace). Popp provides computer readable program code configured to cause a computer to generate a unique name space designation. Popp provides a namespace (group name) that contains a set of named elements, such that the names within the group are resolved to a particular element. See col. 15, line 55 –col. 16, line 9. Each group has a unique name. Popp teaches associating a namespace (group) with the control program code. See col. 4, lines 20-41 and the description of associations, beginning on col. 16, line 48. Popp teaches that the program code uses the unique name space designation to generate a plurality of definitional statements and identify the control with which the unique name space designation is associated. See col. 4, lines 20-26, which show how the plurality of definitional elements can be in a group (namespace) and generated by the group. Also see col. 19, line 60 – col. 20, line 20.

Referring to claims 21 and 28, Popp discloses a GUI system with a processor and method of using a namespace in generating a GUI that models a component of a GUI as a control that is implemented as a program code (col.4, lines 20-63) and generates at least one definitional statement for the component of the GUI using program code. The definitional statement includes at least one attribute for the component of the GUI, which includes a unique namespace designation that is associated with a particular control and capable of identifying the control. See col. 4, line 20-26, which show how the plurality of definitional elements can be in a group (namespace) and generated by the group. Also see col. 19, line 60 – col. 20, line 20.

Referring to claims 15, 22, and 29, Popp teaches that the definitional statement includes a name attribute that specifies a data label and that the method and program code is configured to generate a label to be associated with data, and that label includes the unique name space designation. See col. 16, lines 30-47 and 64-67.

Referring to claims 16 and 23, Popp discloses that the label is associated with the data (col. 16, lines 64-67). Col.20, lines 28-37 describes how

the program code is identified as recipient of the data using the unique name space designation in the label ('FORM.EMPLOYEE'). Also, see col. 12, lines 1-14.

Referring to claims 17 and 25, the plurality of definitional statements in Popp are Hypertext Markup Language (HTML) statements. See col. 3, lines 34-42 and col. 4, lines 48-52.

Referring to claims 18, 24, and 32, the program code (control) is an object-oriented object. See the Element Objects section on col. 11, specifically lines 7-35, which describe how code for definitional statements can utilize object-oriented programming.

Referring to claim 20, Popp discloses that the GUI comprises a plurality of GUI elements, wherein one of the plurality of GUI elements is defined as being within another of the plurality of GUI elements. See col. 4, lines 64-65, which describes that a control object (GUI element) can have subcontrols.

Popp teaches associating a first unique name space designation with program code configured to generate definitional statements for the another if the GUI elements. Popp also teaches associating a second unique name space designation with program code configured to generate definitional statements for the one of the plurality of GUI elements, and the second name space designation includes the first name space designation. See col. 17, lines 1-46. Table 5 shows "WEBPEOPLE" is contained in the "SELECT_FORM" object, and thus includes its name space designation.

Referring to claim 26, Popp discloses the step of generating a design for the GUI that identifies a plurality of GUI components. For example, see col.4, lines 35-41.

Referring to claim 27, Popp discloses a first of the plurality of GUI components in the design is located within a second of the plurality of GUI components, wherein the unique namespace designation is associated with the second of the components and includes the step of generating at least one definitional statement for the first component of the GUI using program code. The definitional statement includes at least one attribute for the first component of the GUI that comprises a first unique namespace designation, which includes the namespace designation associated with the second of the components. See col. 17, lines 1-46. Table 5 shows "WEBPEOPLE" is contained in the "SELECT_FORM" object, and thus includes its name space designation. Also, see col. 16, lines 30-47.

Referring to claim 30, Popp discloses a browser configured to generate a name-value pair, wherein the name portion includes the label. Col.6, lines 40-48 describe the use of a browser, such as Netscape© or Mosaic©, and col. 20, lines 28-38 describe the use of name-value pairs including a label.

Referring to claim 31, Popp teaches a page control configured to examine the name portion of the name-value pair and to direct the name-value pair to the program code based on the namespace designation in the name option. See col. 21, line 60 – col.22, line 50, which describe push and pull methods to send the program code the appropriate value associated with the appropriate name space designation. Also, see col. 26, line 61 – col. 27, line 4.

Applicant respectfully submits that, as amended, independent claims 14, 21, and 28 are allowable because Popp does not teach dynamically generating a unique namespace designation in realtime such that each instance of a GUI component has a unique namespace associated with it.

Popp teaches assigning name attributes to components of a GUI from data stored in a harsh table, for example (Col 15, lines 61-65). Thus, the same name may be used for different instances of the same GUI element. Popp does not teach generating a unique name from the attributes of the element with respect with the GUI.

In contrast, the present invention dynamically generates a unique namespace for each instance of a control element of a GUI at run-time. The namespace is **not** pre-stored in a table database and assigned at invocation like in Popp. Rather the namespace and definitional statements are generated at run-time resulting in a browser independent GUI.

Therefore, since Popp uses pre-generated and stored attribute name instead of a dynamically generated unique names space designation, the present invention is patentably distinct from Popp and is in condition for allowance.

IV. Rejection of Claims Based on 35 U.S.C. § 103(a)

Examiner has rejected claims 19 and 33 under 35 U.S.C. 103(a) as being unpatentable over Popp. Examiner states:

Popp describes how any language could be used as the control, including Java. See col. 10, line 56 – col. 11, line 17 and col. 7, lines 52-58. He does not explicitly teach the use of a Java Bean, however Java Beans are notoriously well known to be used in Java, which Popp teaches as a language of control. The Examiner takes Official Notice of this teaching. It would have been obvious to

use a Java Bean as the control in a GUI, because of their reusability and efficient visual programming.

Applicant respectfully submits that claims 19 and 33, being dependent upon respective allowable base claims, are also allowable for at least the foregoing reasons stated above.

Dependent Claims 15-20, 22-27, and 29-33

Applicant respectfully submits that claims 15-20, 22-27, and 29-33, being dependent upon respective allowable base claims are also allowable for at least the foregoing reasons stated above.

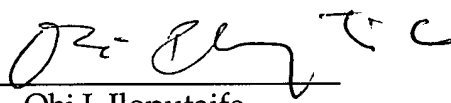
CONCLUSION

For at least the foregoing reasons, Applicant respectfully submits that pending claims 14-33 are patentably distinct from the prior art of record and in condition for allowance. Applicant therefore respectfully requests that pending claims 14-33 be allowed.

Respectfully submitted,

THE HECKER LAW GROUP

Date: 7/9/03

By: 
Obi I. Iloputaife
Reg. No. 45,677

THE HECKER LAW GROUP
1925 Century Park East
Suite 2300
Los Angeles, California 90067
(310) 286-0377

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to Commissioner for Patents, P.O.Box 1450 Alexandria, Virginia 22313-1450, on July 9, 2003



7-9-03

Signature: Mona Kiran Date:



AMENDMENTS TO THE CLAIMS

RECEIVED

JUL 17 2003

Technology Center 2100

What is claimed is:

1.-13. (CANCELLED).

Sub C
14. (CURRENTLY AMENDED) A computer program product comprising:

a computer usable medium having computer readable program code embodied therein for using a name space in generating a graphical user interface (GUI), said computer program product comprising:

computer readable program code configured to cause a computer to associate each element of a GUI with a control mechanism, said GUI having at least one element;

computer readable program code configured to cause a computer to generate a unique name space designation for each instance of said control mechanism at run-time~~computer readable program code, wherein said unique name space designation is not preassigned; and~~

~~computer readable program code configured to cause a computer to associate said unique name space designation with a control implemented with said control program code;~~

~~said computer readable control program code configured to cause a computer to use said unique name space designation to generate a plurality of one or more definitional statements for said GUI and identify said control mechanism with which said unique name space designation is associated.~~

15. (CURRENTLY AMENDED) The computer program product of claim 14 wherein ~~said computer readable configured program code further comprises,~~ further comprising:

computer readable program code configured to cause a computer to generate a label to be associated with data, said label including said unique name space designation.

16. (CURRENTLY AMENDED) The computer program product of claim 15 further comprising:

computer readable program code configured to cause a computer to associate said label with said data;

computer readable program code configured to cause a computer to identify said ~~computer readable control program code control mechanism as~~ recipient of said data using said unique name space designation in said label.

Sub C3
17. (CURRENTLY AMENDED) The computer program product of claim 14 wherein said ~~plurality of one or more~~ definitional statements are Hypertext Markup Language statements.

18. (ORIGINAL) The computer program product of claim 14 wherein said computer readable control program code is an object-oriented object.

19. (ORIGINAL) The computer program product of claim 18 wherein said object-oriented object is a Java Bean.

Sub C3
20. (CURRENTLY AMENDED) The computer program product of claim 14 wherein said ~~GUI comprises a plurality of GUI elements, one a first of said~~

plurality of GUI elements is defined as being within ~~another~~ the influence of a second of said plurality of GUI elements, said computer readable program code configured to cause a computer to associate each element of said GUI with said control mechanism further comprises:

~~computer readable program code configured to cause a computer to associate a first unique name space designation with a first computer readable control program code configured to generate~~ said definitional statement for said ~~another of said GUI element~~ first of said plurality of GUI elements;

~~computer readable program code configured to cause a computer to associate a second unique name space designation with a second computer readable control program code configured to generate~~ said one or more definitional statements for said ~~one second~~ of said plurality of GUI elements, said second unique name space designation including said first unique name space designation.

21. (CURRENTLY AMENDED) In a computer system, a method of using a name space in generating a graphical user interface (GUI) comprising:

modeling a component of a graphical user interface (GUI) as a control, said control implemented as program code; and

dynamically generating at least one definitional statement for each instance of said component of said GUI using said program code, said definitional statement including at least one attribute for instance of said component of said GUI, said at least one attribute including a unique name space designation, wherein said unique name space designation is not preassigned and is derived from said at least one attribute of said instance of said component ~~said unique name space associated with a particular control~~ and is capable of identifying said instance of said control.

22. (CURRENTLY AMENDED) The method of claim 21 wherein said dynamically generating step includes generating a label to be associated with data, said label including said unique name space designation.

23. (PREVIOUSLY ADDED) The method of claim 22 further comprising:

associating said label with said data; and
identifying said program code as recipient of said data using said unique name space designation in said label.

24. (PREVIOUSLY ADDED) The method of claim 21 wherein said program code is an object-oriented object.

25. (PREVIOUSLY ADDED) The method of claim 21 wherein said at least one definitional statement is a Hypertext Markup Language statement.

26. (PREVIOUSLY ADDED) The method of claim 21 further comprising the step of generating a design for said GUI, said design identifying a plurality of GUI components.

27. (CURRENTLY AMENDED) The method of claim 26 wherein a first of said plurality of GUI components in said design is located within a second of said plurality of GUI components, wherein said unique name space designation is associated with said second of said components and including the step of generating at least one definitional statement for said first component of said GUI using said program code, said definitional statement including at least one

attribute for said first component of said GUI, said at least one attribute comprising a first unique name space designation which includes said name space designation associated with said second of said components.

28. (CURRENTLY AMENDED) A graphical user interface (GUI) system comprising:

a processor;

a program code coupled to said processor, said program code comprising a control element and configured to dynamically generate at least one definitional statement for each instance of a component of said a GUI; and

said definitional statement including at least one attribute for said instance of said component of said GUI, said at least one attribute including comprising a unique name space designation, wherein said unique name space designation is not preassigned and is derived from said at least one attribute of said instance of said component and is said unique name space designation associated with a particular control which is implemented by said program code and capable of identifying said particular control instance of said control element.

29. (PREVIOUSLY ADDED) The system of claim 28 wherein said at least one definitional statement includes a name attribute, said name attribute specifying a data label containing said unique name space designation.

30. (PREVIOUSLY ADDED) The system of claim 29 further comprising a browser configured to generate a name-value pair, said name portion of said name-value pair including said label.

31. (PREVIOUSLY ADDED) The system of claim 30 further comprising a page control configured to examine said name portion of said name-value pair and to direct said name-value pair to said program code based on said name space designation in said name portion.

32. (PREVIOUSLY ADDED) The system of claim 28 wherein said control is an object-oriented object.

33. (PREVIOUSLY ADDED) The system of claim 28 wherein said control is a Java Bean.
